
# Smart Legal Contracts

## Summary

# Introduction and findings

Emerging technologies, such as distributed ledgers, are increasingly used to create "smart contracts": computer programs which run automatically, in whole or in part, without the need for human intervention. Smart contracts can perform transactions on decentralised cryptocurrency exchanges, facilitate games and the exchange of collectibles between participants on a distributed ledger, and run online gambling programs.

Smart contracts can also be used to define and perform the obligations of a legally binding contract. It is this specific type of smart contract – a "smart legal contract" – that is the focus of our work.

This document summarises our findings as to how the existing law of contract in England and Wales can apply to smart legal contracts, highlighting any uncertainties or gaps, and identifying such further work as may be required now or in the future. It accompanies a longer paper available at **https://www.lawcom.gov.uk/project/smart-contracts/**.

We have concluded that the current legal framework is clearly able to facilitate and support the use of smart legal contracts. Current legal principles can apply to smart legal contracts in much the same way as they do to traditional contracts, albeit with an incremental and principled development of the common law in specific contexts. In general, difficulties associated with applying the existing law to smart legal contracts are not unique to them, and could equally arise in the context of traditional contracts.

Our findings therefore build on the conclusions of the Legal Statement issued by the UK Jurisdiction Taskforce ("UKJT"), which establishes that the current legal framework is sufficiently robust and adaptable so as to facilitate and support the use of smart legal contracts. The flexibility of our common law means that the jurisdiction of England and Wales provides an ideal platform for business and innovation, without the need for statutory law reform.

The market also has an opportunity to anticipate and cater for potential uncertainties in the legal treatment of smart legal contracts by encouraging parties to include express terms aimed at addressing them. Throughout the paper, we identify particular issues that parties may wish to address in their smart legal contract in order to promote certainty and party autonomy. A non-exhaustive list of these issues is set out in Appendix 1 to this summary. In addition, as smart legal contracts become more prevalent, we anticipate that the market will develop established practices and model clauses that parties can make use of when negotiating and drafting their smart legal contracts. We hope that work in this area could be led by the UKJT or LawtechUK.

# What is a smart legal contract?

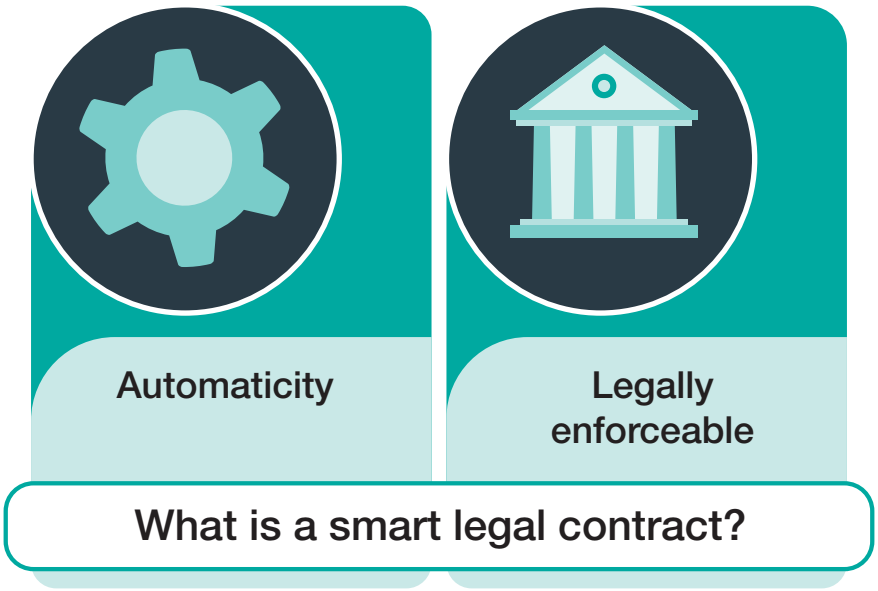
We define a smart legal contract as a legally binding contract in which some or all of the contractual obligations are defined in and/or performed automatically by a computer program. Smart contracts, including smart legal contracts, tend to follow a conditional logic with specific and objective inputs: if "X" occurs, then execute step "Y". Smart legal contracts are expected to increase efficiency and certainty in business, and to reduce the need for the contracting parties to have to trust each other; the trust resides instead in the code.

## Features of a smart legal contract

The two main features of smart legal contracts are:

1. some or all of the contractual obligations under the contract are performed automatically by a computer program ("automaticity"); and

2. the contract is legally enforceable.



### Automaticity

The main benefit of automating a contractual obligation is that it enables contractual performance to occur without the need for human intervention. Contractual obligations which follow a conditional logic (if X, then Y) are good candidates for being coded. For example, a contract for the sale of an asset could be coded so that title to the asset transfers automatically on receipt of a certain amount of money into a particular account.

The use of computer programs to automate the performance of contractual obligations is not new. Automated bank payments (such as direct debits and standing orders) as well as online shopping all involve elements of automation at the instance of one or both of the parties. Owing to their familiarity and extensive use in practice, such contracts are not likely to give rise to novel legal issues. At the other end of the spectrum, a smart legal contract may be drafted primarily or solely in code and deployed on a distributed ledger system. In these cases, where the automation in question takes the contract out of the realm of legal familiarity, novel legal issues may arise.

**Our work is concerned with identifying the characteristics of smart legal contracts that, due to their degree of automation, may require different or novel legal responses.**

### Legally enforceable

The paper focuses on smart contracts that constitute legally binding contracts. We discuss the requirements for formation of a legally enforceable contract later in this summary.

## Distributed Ledger Technology (DLT)

In recent years, DLT has become increasingly sophisticated, to the point where computer programs can be recorded on a distributed ledger and performed by the computers on the network. Smart legal contracts can be deployed on a distributed ledger so that contractual obligations expressed in computer code are performed automatically by the computers on the network. Performance of a smart legal contract is "guaranteed" in the sense that human intervention is not required to facilitate performance.

We do not define smart legal contracts by reference to DLT. Smart legal contracts can be performed automatically by computer programs without the use of DLT. However, DLT systems have distinctive features and benefits which justify a considered analysis. We therefore refer to DLT in the examples provided to draw out the novel issues to which the technology gives rise.

### Distributed Ledger Technology (DLT)

A distributed ledger is a digital store of information or data. It is shared (that is, "distributed") among a network of computers (known as "nodes") and may be available to other participants. DLT is technology that enables the operation and use of a distributed ledger. The distinguishing feature of DLT compared to traditional, centralised databases is that the ledger is not maintained or controlled by a central administrator or entity. This means that network participants do not have to reconcile their local databases with a ledger maintained by a central administrator. Instead, in DLT systems, participants approve and eventually synchronise additions to the ledger through an agreed "consensus mechanism". In general, it requires some or all of the participants to determine the validity of a proposed data entry. The consensus mechanism is typically designed so that, once data is added to the ledger, it cannot (for practical purposes) be amended; it is said to be "immutable".

DLT systems can be permissioned or permissionless and private or public. A permissioned DLT system is generally one in which authorisation to perform a particular activity on the DLT system is required. Permissioned systems tend to be private, meaning that the DLT system is only accessible for use by a limited group of participants. In a permissionless system, no such authorisation to perform activities on the DLT system is required. Permissionless DLT systems tend to be public, meaning that the DLT system is accessible for use by the public. There is not a binary distinction between permissioned and permissionless systems, but rather various degrees and types of permissioning to consider.

## The forms a smart legal contract can take

Although smart legal contracts can take a variety of forms with varying degrees of automation, we consider three broadly-defined forms in the paper. Even though one can classify a smart legal contract according to one of the three forms set out below, it may not always be necessary (or indeed possible) to do so; the form may vary from obligation to obligation.

### Form 1: Natural language contract with automatic performance by code

This is a natural language contract in which some or all of the contractual obligations are performed automatically by the code of a computer program. The code itself does not define any contractual obligations, but is merely a tool employed by one or both of the parties to perform those obligations. This type of smart legal contract can also be referred to as an "external" contract, as the code falls outside the scope of the parties' legally binding agreement. Natural language contracts with automated performance appear to be the most commonly used form of smart legal contract at present. This form of smart contract does not raise any novel legal issues in the context of contract formation or interpretation.

### Form 2: Hybrid contract

A hybrid smart legal contract is a contract in which some contractual obligations are defined in natural language and others are defined in the code of a computer program. Some or all of the contractual obligations are performed automatically by the code. At one end of the spectrum, the terms of a hybrid contract could be primarily written in code with a few natural language terms setting out, for example, the governing law and jurisdiction. At the other end of the spectrum, the terms of a hybrid contract could be primarily written in natural language and include just one or two terms written in code. In addition, the same contractual term(s) can be written in both natural language and in code. The natural language terms can be incorporated in an accompanying natural language agreement, or in natural language comments included in the code.

### Form 3: Contract recorded solely in code

This is a contract in which all the contractual terms are defined in, and performed automatically by, the code of a computer program. No natural language version of the agreement exists. This type of smart legal contract presents the most challenges from a contract law perspective, in terms of determining whether and when a legal contract is formed, and how that contract can be interpreted. Solely code smart legal contracts are likely to be rare in practice, given that commercial contracts are typically too nuanced to be reduced solely to code.

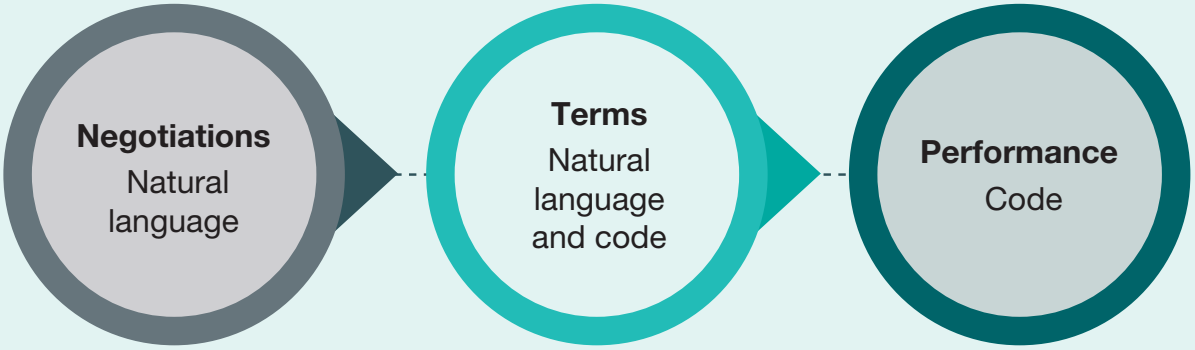
Even though solely code smart legal contracts may not be as frequently encountered in practice as natural language agreements with automated performance or hybrid contracts, we retain the analysis of solely code contracts in the paper. First, including such contracts within the scope of the paper ensures a complete analysis, and accords with the position adopted in the UKJT Legal Statement, which contemplates obligations being defined solely by the code. Second, these types of smart legal contracts give rise to novel legal issues in the context of contract formation and remedies, and their prevalence might increase over time as the underlying technology becomes progressively sophisticated.

**Code may merely perform contractual terms expressed in natural language, or the contractual terms may be defined in, as well as performed by, the code. This will depend on (amongst other things) the smart contract platform, the parties' requirements, and the relevant use case.**

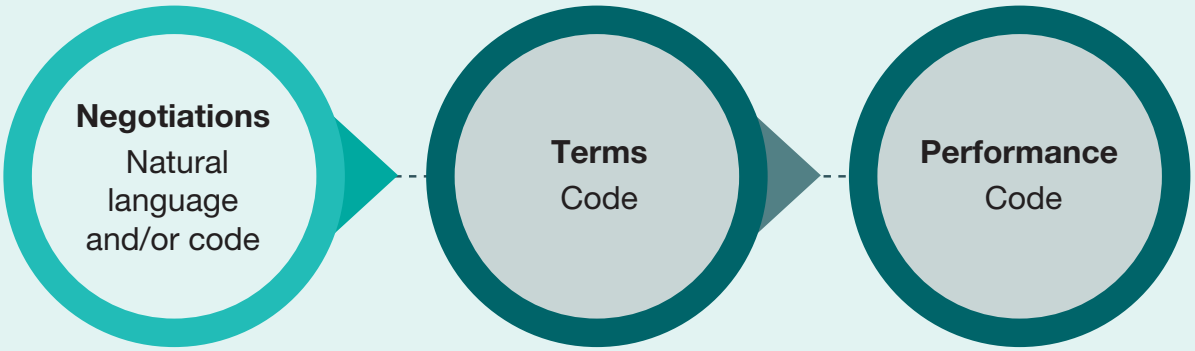
## The three forms of smart legal contract compared

**Form 1:** Natural language contract with automated performance


**Form 2:** Hybrid smart contract




**Form 3:** Solely code contract



## Use cases

Smart legal contracts are increasingly being considered as a means of automating specific processes within conventional contracts, including in the following contexts:

- Facilitating “DeFi” or decentralised finance, which aims to disrupt conventional banking and securitisation arrangements.
- Service level agreement monitoring.
- Real estate transactions.
- Parametric insurance.
- Aviation refuelling.
- Managing supply chains.

# Formation of smart legal contracts

There are several requirements for the formation of a legally binding contract under the law of England and Wales: agreement, consideration, certainty and completeness, intention to create legal relations, and compliance with formalities (including in relation to deeds). Below, we discuss how these requirements might be satisfied in the context of smart legal contracts. We conclude that smart contracts can be used to enter into legally binding contracts under the law of England and Wales, although additional complexities arise in relation to deeds, which are subject to additional formality requirements.

## Agreement

The first requirement for the formation of a legally binding contract is an agreement, comprising an offer to be bound on specified terms, and an acceptance of those terms. We expect that, in most cases, entry into a smart legal contract will typically be preceded by natural language negotiations or other communications between the parties: email correspondence, instructions to coders, exchanges of draft agreements, or oral conversations. In these cases, the question for the court would be the same as in a traditional contracting scenario: did the words and conduct of the parties reveal an offer and acceptance and therefore an agreement?

In principle, it is possible for parties to reach an agreement on a DLT system or other smart contract platform by deploying and interacting with code, without engaging in natural language communications. The relevant question is whether the deployment of the code amounts to an offer, and whether the interaction with the code amounts to an acceptance. In addition, the process of offer and acceptance itself can be undertaken automatically by computer programs, whether on a DLT or non-DLT system, without the need for human intervention. The key question is whether the parties can be said to have "held out" their computer programs for the purpose of reaching an agreement. Ultimately, whether the parties have held out their computer programs in this way is a question of fact which turns on various factors.

The pseudonymous nature of some DLT systems may make it comparatively more common for parties to enter into smart legal contracts without knowing the real identity of their counterparty. There is no requirement under the law of England and Wales for contracting parties to know each other's real identities so the anonymity or pseudonymity of one or both of the parties will not preclude them from reaching an agreement. It may, however, pose challenges later on if there is a dispute and the jurisdiction of the smart legal contract is in question, or if one of the parties seeks to obtain a remedy against the other.

## Consideration

Under the law of England and Wales, contractual promises cannot as a general rule be made "gratuitously" – that is, for nothing in return. The exception is a promise made by deed, which does not require consideration to be legally binding. Our view is that smart legal contracts are unlikely to create particular challenges in identifying the relevant consideration. In most cases, the consideration (such as payment in money or tokens) could be identified from the terms of any natural language agreement, or the operation of the code itself.

## Certainty and completeness

An agreement must be "certain and complete" to constitute a contract. Where a smart legal contract consists of a natural language agreement that is performed automatically by a piece of code, this requirement could be assessed in the ordinary way by examining the provisions of the natural language agreement.

In a hybrid agreement, a potential source of uncertainty is that the natural language and coded terms may conflict with one another. However, conflicts between the code and natural language could be resolved by applying the principles of interpretation, discussed below.

In a solely code agreement, the behaviour of the code may itself provide a strong indication that the agreement is certain and complete, because only complete and syntactically correct code will be executed by a computer. However, in some circumstances, code may execute to give a certain result, but the nature of the legal arrangement is not clear from the code or from the result. The court will, in the first instance, seek to resolve these questions through the process of interpretation. However, if these questions cannot be resolved by interpretation, then there is a risk that the agreement may be found to be uncertain or incomplete.

**A court will only hold the agreement to be uncertain as a last resort.**

## Intention to create legal relations

English law presumes that parties intend to create legal relations when they make an express agreement in a commercial context. An express agreement is an agreement expressed in words, rather than inferred from the parties' conduct.

Where commercial parties conclude a natural language agreement which is then performed automatically by code, the presumption is likely to apply so that there will be little difficulty in proving that the parties intended to create legal relations. However, difficulties may arise if the parties' agreement is made as a result of interaction on a distributed ledger where the agreement is inferred from the parties' conduct rather than as a result of an express agreement. Here, the presumption may not apply so it might be necessary to prove that the parties actually intended to create legal relations.

It is a question of fact as to whether the parties intended to create legal relations. Several factors may be relevant to that assessment, including the nature and purpose of the platform on which the code is deployed and the nature of the transactions executed by the code. English courts have enforced agreements based solely on the parties' conduct where doing so is necessary to give "business reality" to their transaction, or where, in the circumstances, the parties would have expected enforceable obligations to exist.

Some parties may be content to rely on the operation of the code and the relevant DLT protocol to do this and may not want to have recourse to the courts if something goes wrong. Such parties may expressly deny an intention to create legal relations, for example, by including a provision to this effect in the natural language component of their agreement. The courts have given effect to clauses which expressly deny an intention to create legal relations.

**The assessment of whether parties intended to enter into a legally binding agreement could be particularly complicated in the context of an agreement reached on a DLT system or other smart contract platform. Parties who do intend such transactions to create legal relations would be well advised to make this clear in natural language.**

## Formality requirements

The general rule is that contracts need not be made in any particular form. However, some statutes require certain contracts to be made "in writing" and "signed". Deeds are subject to additional formality requirements.

### Can a smart legal contract satisfy an "in writing" requirement?

The Interpretation Act 1978 defines writing as including all modes of "representing and reproducing words in a visible form". If the terms of a smart legal contract are set out in a natural language document, then the smart legal contract would satisfy an "in writing" requirement. The more difficult question is whether contractual terms defined in code could satisfy this requirement. This depends on whether the code is in a form which can be read by a person.

The definition of "writing" in the Interpretation Act 1978 is an inclusive one. It can be interpreted to accommodate new technologies, so long as they involve "representing or reproducing words in a visible form". Source code can be considered a "mode of representing or reproducing words" because it is capable of being read by a person with knowledge of the relevant programming language, and translated into words. Further, as source code can be visibly displayed on a screen or printout, it can be considered a mode of representing or reproducing words "in a visible form".

Whether source code will satisfy a particular statutory "in writing" requirement will depend on Parliament's intention in enacting that specific "in writing" requirement, including other related requirements.

**In our view, source code can constitute "writing" for the purposes of the Interpretation Act 1978. If, however, the terms of a smart legal contract are said to reside in machine code or a lower level of code than source code, then it will be more difficult to argue that the contract is "in writing".**

### Can a smart legal contract be "signed"?

In most cases, contracts governed by the law of England and Wales do not require a signature. Where the law does require a contract to be signed but is silent on the form that the signature must take, the common law generally adopts a pragmatic approach and does not prescribe any particular form or type of signature.

Where the terms of a smart legal contract are recorded in a natural language document, the contract could be signed in the ordinary way. Where a smart legal contract consists solely of code, the parties could sign the contract electronically, for example by using a digital signature to authenticate a piece of code deployed on a DLT system.

**A digital signature is generally capable of satisfying a statutory requirement for a signature. This is the case save where the contrary is provided for in relevant legislation or contractual arrangements, or where case law specific to the document in question leads to a contrary conclusion.**

### Can a smart legal contract be used to create a legally binding deed?

A deed must be signed in the presence of a witness who attests to the signature. If the terms of the deed are recorded in a natural language agreement, performance of which is automated by code, the deed could be signed, witnessed and attested in the ordinary way.

**We do not consider that parties can be confident that the current law supports the creation of deeds which are wholly or partly defined by code. Deeds are documents executed with a high degree of formality and there is some uncertainty as to whether smart contract technology can facilitate compliance with the various formalities that apply to deeds.**

Where the terms of the deed are recorded wholly or partly in code, the question arises as to how a signature authenticating the coded terms of a deed could be witnessed and attested. The current law does not support witnessing other than by the witness being physically present when a deed is signed. If a witness is physically present with an individual when they digitally sign the code, then the witnessing requirement could potentially be satisfied. However, in *R (Mercury Tax Group Ltd) v Her Majesty's Commissioners of Revenue and Customs*, Mr Justice Underhill (as he then was) said that in the case of a deed "the signature and attestation must form part of the same physical document". This requirement may be challenging if existing technology does not allow a witness to record on the smart legal contract that they have observed the execution of that contract.

# Interpretation of smart legal contracts

## The principles of contractual interpretation

A court may be asked to interpret a contract where the parties disagree as to the meaning of the terms of that contract. Such a dispute would usually arise where one party has done something or has failed to do something that another party considers a breach of the contract.

Contractual interpretation is the process by which a court determines the meaning of the language used by the parties in the express terms of a written agreement. The courts of England and Wales take an objective approach to contractual interpretation. The court does not ask what the parties themselves meant by the language they used. Rather, the court asks what the language would have meant to a reasonable person, equipped with all the background knowledge available to the parties at the time the contract was made.

### Are coded terms amenable to contractual interpretation?

Given that computers do not "interpret", but merely execute coded instructions, it may be tempting to conclude that coded terms are not susceptible to the exercise of contractual interpretation at all, or that the principles of interpretation are redundant when interpreting coded terms.

We suggest, however, that coded terms can (and should) be susceptible to contractual interpretation. We do not think that the code simply means what the code does when it is executed, or that it has no meaning, and only an effect. In our view, there can be a divergence between what the code "means", and what it does when it is executed, which entails a distinction between meaning and effect. The interesting question that then arises is: how does one ascertain the meaning of the code?

## Applying the principles of interpretation to smart legal contracts

Novel interpretation issues are unlikely to arise where the terms of a smart legal contract are recorded exclusively in a natural language contract, and a piece of code merely automates performance of those terms. The natural language contract will be treated as containing the terms agreed to by the parties, and it will be those terms that the court will be called upon to interpret.

However, where the terms of a smart legal contract are recorded partly or solely in code, this potentially poses difficulties for contractual interpretation. The principles of interpretation have been developed in response to courts seeking to interpret natural language terms. This raises a question as to how existing principles can be used to interpret coded terms where disputes about the "meaning" of such terms arise.

### The appropriate test for interpreting coded terms

There appear to be two alternative avenues for ascertaining the meaning of a coded term of a smart legal contract, other than asking what a reasonable person would understand the coded term to mean. One approach would be to ask how the coded term would be understood by a functioning computer. However, this would entail reducing interpretation of the code to simply

observing its performance, or output, since the language of code can only have one "meaning" to a computer. We think the more appropriate test would be to ask what a person with knowledge and understanding of code would understand the coded term to mean – that is, a "reasonable coder".

An expert coder could assist the court by translating the code in the same way as any other contract written in a language unfamiliar to the court. Nonetheless, a court may not be able effectively to interpret that natural language translation in the same way as it could with the translation of a foreign language. This could be because the court is unfamiliar with the way instructions in code are interpreted by a computer, or with the way a coder might arrange instructions in order to elicit a particular outcome from the running of a code. Instead, the expert coder will need to explain the effect of certain combinations of words, and give their reasoned opinion as to what the code appeared to instruct the computer to do.

We acknowledge that adopting this test entails a nuanced development of the existing principles of interpretation. However, we think that such a development is necessary and justified in order to take account of the unique nature of contracts written in coded terms.

**The "reasonable coder" test has the benefit of providing an insight into what the parties intended the code to do, regardless of the computer's ultimate performance. It is more consistent with the existing approach to contractual interpretation than one that asks what the code meant to a functioning computer.**

## Natural language aids to interpreting coded terms

### Business process document

The parties to a smart legal contract may prepare a business process document or term sheet which sets out in detail the terms of the transaction. The document can then be handed over to a coder to translate into code, which constitutes the smart legal contract, and is signed by the parties. The question that arises is whether the document can be relied upon in interpreting the coded terms. In such a case, much will depend on whether the business process document has been agreed to by both parties, and is a legally binding contract. If it is, then the business process document can be considered an antecedent (or prior) agreement to the solely code contract.

**As a rule of interpretation, an antecedent agreement may be relied upon in interpreting a later agreement. However, the usefulness of such a prior contract in interpreting the terms of a later contract will depend on the facts of the case.**

### Natural language explanation of the code

Natural language can be used in various ways to aid the court in understanding and interpreting the coded terms of a smart legal contract. For example, the parties can set out expressly, in natural language, how they intend the code to operate. There are various forms that a natural language explanation of the code can take. In the context of a hybrid smart legal contract, for example, the natural language component could include terms setting out in detail how the code is intended to operate, or simply be a broad statement of intent.

The question arises as to how and when such natural language explanation could be considered by a court faced with interpreting coded terms. This will depend on the nature and construction of the natural language explanation, and whether it forms part of the parties' contract. If it forms part of the contract itself, there is no issue in admitting such explanation as an aid to interpreting the coded terms. If it does not form part of the contract, it may still be relevant in interpreting the coded terms on the basis that it forms part of the admissible background. In this case, however, the admissibility of such material is subject to the limitations associated with admitting background material, which includes that such material cannot be used to ascribe to the words of the contract a meaning that they cannot legitimately bear.

To ensure a natural language explanation of the code is taken into account when interpreting coded terms, the parties could expressly state that such explanation forms part of their legally binding agreement, or (where the natural language explanation is contained in a separate document) expressly incorporate by reference the terms of such a document into their coded agreement.

### Natural language comments in source code

Where the comments in code constitute contractual terms, such comments will be relevant to the interpretation of the smart legal contract as a whole as they form part of the contract. If a dispute were to arise about the meaning of the coded terms of the contract in particular, the meaning of the terms embodied in the comments in the code would be relevant to the court's interpretation of the coded terms in dispute.

If the comments in the code do not constitute contractual terms, we think such comments can still be admissible as a useful aid to the interpretation of the coded terms of a smart legal contract. For example, where the comments in the code explain what a single line of code will do, we think an analogy can be drawn with headings in traditional contracts. Unless the contract stipulates otherwise, headings can generally be considered in construing the meaning of a particular clause, but they cannot override clear language, or create an ambiguity where, but for the heading, none would otherwise exist.

**To ensure that natural language comments in the code are taken into account when interpreting coded terms the parties could, for example, expressly state that such comments form part of their legally binding agreement.**

### Evidence of the parties' pre-contractual negotiations

Evidence of the parties' prior negotiations as to the meaning of the words used is not admissible. Admitting such evidence is generally thought to be unhelpful because what the parties did and said in their negotiations may not reflect the final position they took when they entered into the contract.

We do not think it is necessary or desirable to admit evidence of the parties' pre-contractual negotiations to assist in the interpretation of the coded terms of a smart legal contract. We do not think there is sufficient justification for proposing a special rule for smart legal contracts in this context.

**Given that admitting pre-contractual negotiations as an aid to interpretation would involve a reform of the law, we do not think there is sufficient justification for proposing it. Furthermore, there are various natural language aids to interpreting coded terms that can be relied on.**

# Remedies and smart legal contracts

Various problems can arise in the lifecycle of a contract, and in response to these problems the law provides a range of remedies. We discuss the problems that might arise in the context of smart legal contracts, the remedies that the parties might seek, and how a court might award those remedies in practice. We think that smart legal contracts may increase instances of defective performance, given the scope for the code to perform in ways the parties did not intend.

## Rectification

In principle, the remedy of rectification could be available to amend the coded terms of a smart legal contract where, for example, errors in translation result in the code failing to give effect to the parties' actual common intention at the time of contracting.

A court may however face practical difficulties in rectifying coded terms where, for example, the coded terms are recorded on an immutable distributed ledger. Where the relevant smart legal contract is of such a nature that it cannot be rectified, a workaround may be needed. For example, where the code is deployed on a permissionless blockchain, the court could identify the error which needs to be rectified, and ask the parties to agree upon a revised piece of code. The court would then order the parties to deploy the revised code on the blockchain.

Another practical difficulty with rectifying coded terms is that a party may only discover the error in the code after the code has executed. In these circumstances rectification may not, in itself, provide an effective remedy for a claimant, who will want to reverse the effects of the code's performance. Even so, we think that rectification may be of value in cases where the code has already fully performed insofar as it provides a basis for the award of other remedies, such as breach of contract. In addition, we think that rectification is likely to be relevant where the parties have entered into an ongoing contract, or one that requires continuous performance, and where the code may have partially (rather than fully) performed.

Rectification may be particularly relevant in a smart legal contract context where the parties enlist the services of a coder to "translate" their bargain into code.

**The extent of any practical difficulties in rectifying coded terms will likely depend on the technical specifications of the particular smart contract platform, such as whether it has relevant built-in functionality to rectify the coded terms.**

## Vitiating factors

The law recognises various "vitiating factors" that render a contract defective. These include mistake, misrepresentation, duress and undue influence. A vitiating factor may render the contract "void" (of no effect from the start) or "voidable" (liable to be set aside from the start).

## Mistake

A contract can be rendered void if one or both parties laboured under a mistake when entering into the contract. A "mistake" can be described as an erroneous belief or assumption about a matter of fact or law. Mistake is a common law doctrine which has a narrow scope under the law of England and Wales.

### Common mistake

The doctrine of common mistake concerns the situation where the parties enter into a contract under a mistaken belief or assumption about a matter of fact or law relating to the subject matter of the contract, or the facts surrounding the formation of the contract.

When parties enter into a smart legal contract, they may hold certain beliefs or assumptions about how the code will perform. Where the code executes in a way contrary to those beliefs or assumptions, the question that arises is whether the smart legal contract can be vitiated on the ground of common mistake.

In the smart legal contract context, there are increased opportunities for parties to be mistaken about something fundamental or material to the performance of the contract. We do not, however, think this necessitates expanding the scope of the doctrine of common mistake. The ability of the parties to establish a common mistake will depend on the facts of the particular case. In our view, the same principles of common mistake should continue to apply to smart legal contracts as they do traditional contracts. In terms of determining whether a common mistake was made when entering into a smart legal contract, the existing law suffices.

**Where the parties hold beliefs or make assumptions about how the code will perform, the parties would be well advised to allocate the risks of any potential mistakes by appropriate drafting.**

### Unilateral mistake

The doctrine of unilateral mistake concerns the situation where only one of the parties is mistaken at the time the contract is made. If it can be shown that (at the time of entry into the contract) a party was mistaken as to a term of the contract, and the other party knew of this mistake, the contract is void. One situation which may pose challenges in applying the principles of unilateral mistake is where the coded terms of a smart legal contract are offered and accepted by computer programs on behalf of the parties.

We do not think a fundamental change to the existing principles of unilateral mistake in the context of smart legal contracts concluded by computer programs is needed. However, we think that some change is required; a deterministic computer program will always operate within the parameters set by the programmer, and will only do what it is programmed to do.

In this regard, we think that the test for determining whether a non-mistaken party has knowledge of the mistaken party's mistake, where the smart legal contract is concluded through the autonomous operation of computer programs, requires adaptation. This entails, amongst other things, formulating a test that addresses whose knowledge of the mistake is relevant, the time frame for assessing that person's knowledge, and the type of knowledge that is required.

**Any adaptation of the test that the courts adopt should be incremental, and sensitive to the unique nature of smart legal contracts concluded through the autonomous operation of computer programs.**

## Misrepresentation

A contract is voidable if a party is induced to enter into the contract by a misrepresentation made by the other party. We do not anticipate that smart legal contracts will give rise to novel legal issues in this area. Like traditional contracts, entry into a smart legal contract will typically be preceded by a period of negotiation between the parties. Whether a party made a misrepresentation, by their words or conduct, which induced the other party to enter into the smart legal contract, can be determined by applying the existing law.

If the parties have included clauses in the natural language element of their smart legal contract which seek to exclude or limit liability for misrepresentation, the existing law can be applied in the conventional way. The question would be whether those clauses were "fair and reasonable" under the Unfair Contract Terms Act 1977 or, in the case of consumer contracts, "unfair" under the Consumer Rights Act 2015.

## Duress and undue influence

A contract is voidable if a party is induced to enter into the contract by an illegitimate threat made by the other party, or where a party enters into the contract under the undue influence of the other party. We do not anticipate that smart legal contracts will give rise to novel legal issues in these areas. As in the case of traditional contracts, the question will be whether the claimant entered into the smart legal contract because of illegitimate threats made by the other party or in the context of a relationship of undue influence.

## Remedies where the contract has been vitiated

We anticipate that restitutionary remedies may be particularly relevant in the context of smart legal contracts. In a traditional contracting scenario, parties are likely to cease performing the contract when they discover the factor rendering the contract void or voidable. However, in a smart legal contract context, some or all of the terms are performed automatically by code. Depending on the nature of the platform and the code in question, there may be no mechanism for the parties to stop the execution of the code.

We do not anticipate that English courts will encounter novel legal issues or practical difficulties in awarding restitutionary remedies if a smart legal contract is void. The fact the contract is void may provide a foundation for a claim in unjust enrichment, the remedy for which is restitution. The court could identify the benefits transferred by the code under the void contract, value those benefits in money, and then order the parties to make restitution to each other to the value of those benefits.

Where a smart legal contract is voidable, it may be set aside so long as the parties can be restored to their pre-contractual positions. English courts do not insist on literal restoration; instead the court aims to achieve "practical justice" between the parties. We think that the existing legal principles of rescission can be applied to smart legal contracts without difficulties.

However, if the smart legal contract has been partly or wholly performed by code, the question arises as to how the parties can be returned to their pre-contractual positions. If the code of a smart legal contract has performed transactions on a blockchain, those transactions may not be capable of literally being unwound because they are recorded on an immutable distributed ledger.

Even so, there may be other ways in which the court could achieve "practical justice" between the parties. For example, the court could order the parties to enter into an "equal and opposite" second transaction on the blockchain. The first transaction would remain on the blockchain, but its effects would be reversed by the second transaction. The precise nature of the order fashioned by the court will likely depend on various factors, including the type of smart contract platform, and whether the code has already fully performed.

**Even if the remedy is not rescission in a strict legal sense, in practical terms the result is the same, and this may well be sufficient in the majority of cases.**

## Breach of contract

### Breach of contract where natural language obligations are performed by code

If the code is merely a tool used by the parties to perform their obligations under a natural language contract, then a party may be liable for breach of the contract if the code fails to perform those obligations correctly. The party in breach may have to pay damages to place the other party in the position they would have been in had the code executed correctly. We consider that the existing principles of awarding damages for breach of contract should not create difficulties where the terms of a natural language contract are performed automatically by computer code.

In addition, we see no reason why a natural language contract, the performance of which is automated by code, could not be terminated for breach. However, as a practical matter, the party who elects to terminate the smart legal contract may not have the power to terminate performance of the code, particularly if the code is recorded on an immutable distributed ledger.

We understand that, in some cases, it may be possible to design the coded element of a smart legal contract so that the parties themselves can initiate a "kill" or "self-destruct" mechanism, which terminates the performance of the code. The innocent party who elects to terminate the contract could potentially initiate this mechanism to ensure that the performance of the code is also ended when the natural language contract is terminated.

**Although smart legal contracts are likely to reduce the incidence of non-performance, that is not necessarily the same as reducing or removing instances of breach of contract for defective performance.**

### Breach of contract where contractual obligations are defined in code

Where the code is not merely a tool for performing the parties' contractual obligations, but instead defines those obligations, it may be more difficult to establish a breach of contract. This is because establishing a breach of the coded terms of a smart legal contract is likely to first require the parties to establish the meaning of those terms, and interpreting coded terms is itself a difficult exercise. We consider that any difficulties that may arise in this context will primarily be in relation to interpreting coded terms, rather than in applying the principles of breach of contract to those terms once their meaning has been established.

Once the meaning of the coded terms has been settled, the court will be able to apply existing principles to determine whether a breach of the coded terms has occurred and, if so, to award the appropriate remedy.

## Frustration

The doctrine of frustration concerns the situation where the parties have entered into a contract, but by reason of a subsequent event, performance has become physically or legally impossible or something "radically different" from what was contemplated by the contract.

We consider that the existing principles of frustration can accommodate smart legal contracts, even though they may give rise to new types of frustrating events. In addition, we think that frustration may assume greater significance in the smart legal contract context, because of the range of factors external to the parties' control that can render performance of the code impossible or "radically different" from what was contemplated by the contract.

In principle, a smart legal contract could be frustrated where, by reason of a subsequent event, performance of the code becomes physically impossible (for example, if the platform on which the code is deployed is shut down due to some unforeseen event). In other cases, the subsequent event might not prevent the code's performance, but might cause the code to execute in a way "radically different" from that contemplated by the contract.

Where the smart legal contract is frustrated due to an external event rendering performance of the code legally impossible or radically different from what was contemplated by the contract, any future performance of the code will need to be terminated. In these cases, practical difficulties might arise, similar to those encountered in relation to other remedies, such as in termination for breach of contract.

Parties to a smart legal contract would be well advised to draft detailed provisions that deal with the risk of external events beyond the parties' control affecting performance of the code.

## Illegality

A concern that is sometimes expressed about smart contracts is that they may facilitate illegal activity. Some DLT systems enable the parties to transact using pseudonyms, without disclosing their real identities. Further, DLT enables participants to transact directly with one another, without the need for intermediaries, such as banks, who would traditionally play a role in detecting illegal activity. Finally, the immutability of data on a DLT system may make it difficult for authorities to halt the code's performance once illegal activity is detected.

If the purpose or performance of a contract involves conduct that is illegal, then the contract may not be enforced by a court. A rationale for this principle (known as the "illegality doctrine") is that it would be contrary to the public interest to enforce a claim if doing so would harm the integrity of the legal system. We consider that the existing principles of the illegality doctrine can apply to smart legal contracts, and that no specific modification to existing principles is necessary. In terms of halting performance of the code after a finding of illegality, we think this is a practical difficulty rather than a difficulty in applying existing legal principles. However, since the coded element of a smart legal contract performs automatically, it is perhaps unlikely that a party would ask a court to enforce the smart legal contract. It seems more likely that a party might bring a restitutionary claim to recover money or property transferred under a smart legal contract tainted by illegality.

# Consumers and smart legal contracts

Most of the legal issues raised in the paper apply to all smart legal contracts, whether business to business commercial contracts, peer to peer arrangements or business to consumer ("B2C") contracts.

Any business decision about whether to develop and use a B2C smart legal contract is likely to depend on various factors. We note that efficiency considerations, the ability to access reliable data, and the availability of standards and open-source tools for the creation of smart legal contracts are factors which are likely to be relevant in determining whether to use a smart legal contract. However, a unique consideration that arises in the context of B2C smart legal contracts is whether they can be used consistently with consumer protection laws.

## Consumer protection and smart legal contracts

### Transparency and fairness requirements

Under the law of England and Wales, specific consumer protections apply to "consumer contracts", which are contracts entered into between a trader and a consumer. These consumer protections are principally set out in the Consumer Rights Act 2015 ("CRA 2015") and in various regulations implementing EU Directives.

Under section 68 of the CRA 2015, a trader must ensure that the written terms of a consumer contract are transparent. In order to be transparent, the terms must be expressed in plain and intelligible language and be legible. The coded terms of a B2C smart legal contract may not be "transparent" to a non-code literate consumer in the absence of a natural language explanation of those terms. This is because the average consumer is unlikely to be able to read and understand code. From the consumer's perspective, code is unlikely to be readable, comprehensible or informative.

Terms of a B2C smart legal contract which are drafted in code and not accompanied by a natural language explanation may be more susceptible to a finding of unfairness. Such terms may be considered contrary to good faith on the basis that they are not expressed fully, clearly and legibly, and serve to take advantage of the consumer's lack of familiarity with code.

It is conceivable that a trader may seek to include statements in their pre-contractual literature that a transaction with a consumer is not intended to give rise to legally binding relations. In the consumer context, we consider that such statements may be at risk of being unfair consumer notices under section 62(6) of the CRA 2015.

**Traders who seek to offer B2C smart legal contracts which contain coded terms would be well advised to provide clear and informative pre-contractual literature to the consumer, explaining those terms and how they operate.**

### Consumers' right to treat the contract as at an end

A consumer has various statutory rights to treat a consumer contract as at an end in certain circumstances. As smart legal contracts perform automatically and may therefore not be easy to halt, it may be difficult practically for the consumer to exercise these rights. Traders would be well advised to design the B2C smart legal contract so that, where a consumer wishes to exercise their right to treat the contract as at an end, they have the practical means of doing so.

### Are additional protections necessary?

We do not consider that it is necessary at the present time to introduce a separate legal requirement that traders provide a natural language explanation of coded terms to consumers. In our view, the existing law effectively places traders under such an obligation.

**The adequacy of existing consumer protection laws should be kept under review as B2C smart legal contracts become increasingly sophisticated and prevalent.**

# Jurisdiction and smart legal contracts

When problems arise in relation to cross-border contracts, and in the absence of a jurisdiction (or choice of court) agreement between the parties, the rules of private international law determine which national courts have jurisdiction to hear and adjudicate upon the parties' claims. There are a number of factors that may be relevant in determining whether the courts of England and Wales have jurisdiction to hear a cross-border dispute in relation to a smart legal contract.

## Contracting parties and the circumstances of contract formation

One factor which may be relevant in determining jurisdiction is the physical location of the defendant. The pseudonymous nature of some DLT systems may make it comparatively more common for parties to enter into smart legal contracts without knowing the real identity of their counterparties. This poses obvious challenges for determining whether a court's jurisdiction can be based on the defendant's presence within England and Wales, or whether a claimant must instead obtain the court's permission to serve the claim on a defendant located outside of England and Wales.

### The contract's place of formation

The place of formation of a smart legal contract may also be relevant to determining jurisdiction. Conventionally, a contract is formed at the moment when, and in the place where, the offeree's acceptance of an offer is communicated to the offeror.

Identifying the place of formation may be more challenging in the smart legal contract context. The challenges stem from a variety of factors. These include the fact that there may be little or no natural language interaction between the parties, and the fact that different elements of the smart legal contract may be dispersed across a wide range of different jurisdictions.

We think that the more elements involved in the formation of a smart legal contract and, as a consequence, the more legal systems potentially engaged, the more challenging (and perhaps artificial) it will be to identify a particular place of formation.

For smart legal contracts, the analysis of the place of formation will ultimately depend upon the form that the smart legal contract takes. The most challenging analysis will involve solely code smart legal contracts. In particular, unilateral solely code contracts and contracts formed by the autonomous interaction of computer programs.

For both of these types of smart legal contract, the appropriate analysis may be that the parties have implicitly waived any requirement than an acceptance be communicated. Accordingly, parties to such contracts will not necessarily be able to rely on the conventional rule for identifying a contract's place of formation.

**To mitigate the uncertainties in relation to place of formation, parties would be well advised to include a jurisdiction clause in their smart legal contract.**

### The relevance of agents

Under the common law rules, a court will have jurisdiction if a contract was made by or through an agent trading or residing in England or Wales. Smart legal contracts raise interesting questions about whether third-party coders, and even computer programs themselves, can be regarded as agents of one or both of the contracting parties.

We think that a third-party coder could be the agent of one or both of the contracting parties, depending on the precise relationship in a particular case. In contrast, we consider that a computer program would not be regarded as an agent.

## Applicable law

The law applicable to a contract can be:

- a basis for establishing the court's jurisdiction; and/or
- a factor in determining the comparative appropriateness of a particular court.

Parties can agree upon a contract's applicable law, and express that agreement in their contract. We recognise that, at least in theory, parties may intend to create legal relations, but purport to choose that their agreement be governed solely by the protocol of a particular platform. However, we do not think that this is a choice open to parties under the Rome I Regulation. Additionally, we think that there are compelling arguments for this remaining the case. Instead, a better mechanism for reflecting a platform's protocol in an agreement may be to incorporate the protocol rules as terms of the smart legal contract.

**It would be advisable for parties entering into a smart legal contract to designate the law applicable to their contractual relationship. Such a choice is likely to provide parties with clarity as to the content of their obligations, and the consequences of any wrongdoing. It may also provide certainty in relation to the parties' intention to create legal relations.**

In the absence of a choice of law clause, the law applicable to a contract will be determined by its connections to different legal systems. A significant challenge for private international law in the smart legal contract context will be the task of identifying significant connecting factors where the contract engages a great variety of different legal systems. Nevertheless, we think that there is broad continuity between the connecting factors that apply to non-smart legal contracts, and the connecting factors that apply to smart legal contracts. Courts will have to conduct the same type of fact-sensitive enquiry, albeit certain novel factors may have to be taken into consideration in the smart legal contract context. These could include the location of any private key, or the domicile of any central administrator, if the relevant ledger is permissioned.

## Performance, breach, acts and enrichment

Jurisdiction rules are often based on a legally significant connection between a contractual dispute and a particular legal system. Such a connection can be found in the place where the contract was breached, the place of acts giving rise to an alleged liability to make restitution, or the place where a party is enriched.

**Smart legal contracts may present unique challenges when seeking to identify the geographical location of breaches, actions, and enrichments, particularly where the obligations under a smart legal contract concern a digital asset, rather than a physical asset with a clear real-world location.**

## Jurisdiction rules for special types of contracts

Certain types of contracts engage specialised jurisdiction rules. These rules are usually tailored towards providing the party in the weaker position either more choice of where they can sue, or more protection in relation to where they can be sued.

We think that the smart nature of any consumer or employment contract is unlikely to affect the operation of these rules.

## Comparative appropriateness

Once a court has established a basis for jurisdiction under the common law rules, the court must then determine whether it is appropriate for it to accept jurisdiction, and adjudicate upon the contractual dispute before it.

The evaluation of the comparative appropriateness of England and Wales as a forum for the dispute is a practical and fact-sensitive enquiry. A range of factors may be relevant, including:

- the place of formation of the smart legal contract;
- the location of the contractual subject matter;
- the place of performance or breach;
- the location of the nodes participating in the distributed ledger;
- the location and/or domicile of the contracting parties, and any other relevant witnesses and evidence;
- the law applicable to the smart legal contract, and the complexities of the legal issues raised;
- the contracting parties' centres of main interest; and
- the location of the smart legal contract platform.

## Digital location

Two issues were identified as the most problematic matters in applying existing rules of private international law to smart legal contracts. These were determining the location of digital assets, and determining the location of particular actions that "take place" on a distributed ledger.

**We agree that the problem of digital location – that is, the difficulty of ascribing real-world locations to digital objects and digital actions – is amongst the most significant challenges that private international law will have to overcome in relation to emerging technology, including smart legal contracts. We will consider these issues and potentially propose solutions as part of our future project on conflicts of laws in the context of emerging technology.**

# Appendix 1: A non-exhaustive list of issues parties may wish to provide for in their smart legal contract

We set out below a non-exhaustive list of issues that parties may wish to consider (and possibly provide for) in their smart legal contract. This list is not exhaustive, but is intended merely to assist parties who wish to enter into smart legal contracts by drawing their attention to certain issues that may require upfront consideration, and possible treatment, in their contract. It is intended to provide parties with an insight into the sorts of issues that could lead to disputes if the parties have not properly provided for such matters in their contract. Dealing directly with these issues in contractual terms should reduce uncertainties regarding the legal treatment of the parties' smart legal contract, and reduce the scope for potential disputes.

1. Before entering into a smart legal contract, parties should consider engaging in a rigorous planning phase to understand business requirements, and the objectives of the smart legal contract.

2. Parties should give thought as to the form the smart legal contract will take, and whether the form will vary between individual obligations.

3. Parties would be well advised to make clear the role of the code in their smart legal contract and, in particular, to specify if the code is intended both to define contractual obligations as well as to perform them, or only to perform them.

4. Parties should consider the relationship between any natural language and coded terms. In particular, where the same term is expressed in both natural language and in code, parties would be well advised to make clear which term takes precedence in the event of a conflict.

5. Smart legal contracts and associated technologies require parties to consider a broader range of factors before contracting than they would otherwise consider before concluding a traditional contract. Parties would be well advised to allocate risk in relation to, and to provide for (amongst other things):

    a. a malfunctioning oracle or inaccurate data inputs;

    b. external events beyond the parties' control which may affect performance of the code, such as system upgrades;

    c. bugs and coding errors in the code; and

    d. any potential mistakes that may arise due to the parties holding certain beliefs or assumptions about how the code will perform.

6. Parties would be well advised to make clear the role of any non-executable comments in the code and, in particular, if such comments constitute contractual terms.

7. Where the smart legal contract contains coded terms, parties would be well advised to provide a natural language explanation of the workings of the code, and to make clear that such explanation forms part of their contract. An understanding of the parties' intentions will be relevant in the event that the code performs in a way not expected by the parties.

8. Parties who intend their transactions on a DLT system or other smart contract platform to create legal relations would be well advised to make this clear in natural language, either in a separate agreement or by way of comments in the code.

9. Parties would be well advised to consider designing the coded element of their smart legal contract such that performance of the code can be terminated if necessary. Thought will have to be given as to how best to structure this functionality so as to avoid any associated risks of abuse by one of the parties. Similarly, to avoid a scenario where the code performs pending the outcome of a dispute, parties would be well advised to provide a mechanism for suspension of performance of the code in their smart legal contract.

10. Parties would be well advised to include choice of court and choice of law clauses in their smart legal contract, either in a separate natural language agreement, or by way of comments in the code.